Technology
Arts Sciences
TH Köln

Identifying Key Opinion Leaders in Social Networks

An Approach to use Instagram Data to Rate and Identify Key Opinion Leader for a Specific Business Field

Master thesis

by

Christopher Egger

submitted to obtain the degree of

Master of Science (M.Sc.)

at

TH Köln - University of Applied Sciences
Institute of Informatics

Course of Studies

Web Science

First supervisor: Prof. Dr. Kristian Fischer
TH Köln - University of Applied Sciences

Second supervisor: Prof. Dr. Gerhard Hartmann
TH Köln - University of Applied Sciences

Cologne, April 2016

Contact details: Christopher Egger
Tilsiter Str. 5
50735 Cologne
christopher.egger@smail.th-koeln.de

Prof. Dr. Kristian Fischer
TH Köln - University of Applied Sciences
Institute of Informatics
Steinmüllerallee 1
51643 Gummersbach
kristian.fischer@th-koeln.de

Prof. Dr. Gerhard Hartmann
TH Köln - University of Applied Sciences
Institute of Informatics
Steinmüllerallee 1
51643 Gummersbach
gerhard.hartmann@th-koeln.de

Abstract

This thesis focuses on the identification of influential users, also known as key opinion leaders, within the social network Instagram. Instagram is a very popular platform to share images with the option to categorise the images by certain tags. It is possible to collect public data from Instagram via the open API of the platform.
This thesis presents a concept to create an automated crawler for this API and collect data into a database in order to apply algorithms from graph theory to identify opinion leaders afterwards. The sample topic for this thesis has been veganfood and all associated posts from Instagram have been crawled.
After the user data has been crawled a graph has been created to do further research with common social network analysis tools. The graph contained a total set of more than 26,000 nodes.

To identify opinion leaders from this graph, five different metrics have been applied, in particular PageRank, Betweenness centrality, Closeness Centrality, Degree and Eigenvector centrality. After applying the different algorithms the results have been evaluated and additionally an marketing expert with focus on social media analysed the results.
This project was able to figured out that it is possible to find opinion leaders by using the PageRank algorithm and that those opinion leaders have a very good value of engagement. This indicates that they show a high interaction with other users on their posts. In conclusion the additional research options are discussed to provide a future outlook.

Contents

1. Introduction

The modern online world changed the way people connect with each other and how we interact with certain topics, products and so on. Social networks services like Facebook, Instagram, YouTube, Twitter and many more are rising day by day with a wider range of everyday users as well as opinion leaders in specific topics. Many companies have already seen these people as a huge potential to enhance their own position in the market and improving their business communications. With these technologies businesses are closer to their customers than before and communications are more like a dialog instead of a monolog.

In network theory there are already studies which state that there is a possibility that people can be connected with each other over six edges, also known as the "small world phenomenon". Using the technology of modern social networks we can make use of this phenomenon to connect to more people than ever before.
There are studies in the field of sociology as well, which have proven that there are people within networks that tend to have more power over others and take a pivotal position in the network. Of course there are celebrities from the field of music or film making which can influence others to buy products. But the modern world especially the digital natives are more likely to see those pivotal people in their networks as influencer.
In the field of marketing these influential people are called opinion leaders or key opinion leaders, they represent a small amount of people with a very deep knowledge in a certain topic or with an superior skill set in a specific field. Thus key opinion leaders are most likely experts in their field they represent and people which others trust in the part of the content they distribute. But opinion leaders can also vary from field to field and what they represent, for example some professional vlogger on YouTube, with the main goal to create daily video content can also be representative for a technological opinion leadership, by things they use or buy. Other users which follow those people are more likely to buy those things.

1.1. Problem description

One of the main methods for running a successful business is customer acquisition and gain increasing customer prospects and inquiries. This can be achieved by various well established marketing techniques.
Recent marketing strategies make use of the increasing usage of social media networks and involve strong connected persons with a high value to the businesses target group. Realising continuous improvement in customer acquisition can become a complex and time consuming workflow. The possibilities of social media networks provide a promising opportunity for companies to take advantage of these active and fast moving online communities. Target groups can be discovered, clustered and the marketing can be based on a high amount of user characteristics that are available online. Customers can be grouped based on various traits like demographics, or behavioural variables which result in a more granular and more target oriented marketing possibilities.
Whereas these modern techniques can be conducted in a highly automated way by defining and applying suitable algorithms, they are still directly focusing the potential customers advertisement. These advertisements are more user related and reduce the risk of investing into marketing that reaches people outside of the desired target group. However the user him/herself still recognises these approaches as direct advertisement. In order to take these modern approaches one step further the phenomenon of key opinion leaders in social networks can be used. By engaging successful and active users with the product and getting them to directly or indirectly promote it, a company has the opportunity to reach an even greater attention.

Certain topics for example like food or vegan food have a community where some people within that community or network have an important position and a high reputation from community members. Some persons are followed by many others and the opinions of these persons have a high credibility for all others.

Businesses like to use such powerful persons to get a better communication to their customers or distribute their opinions within those communities. Manually finding opinion leaders in social networks can be a very time intensive and complicated work, this includes analysing the network, finding persons with a high popularity and deciding if the person fits to the business. Moreover the risk of missing important opinion leaders when finding them manually is high. With an automated approach it is possible to cover a larger scope and reduce the risk of a too small user base.

1.2. Goal

This thesis will be elaborated within the context of an real world environment, in particular in a marketing agency. Finding key opinion leaders is a strong emerging issue and it is a very time consuming and expensive task when it comes to human resources.
The goal of this study is to develop a concept for an automated approach of key opinion leader mining and identifying. If it is possible to receive valuable results from an automated approach this could save a lot of resources and the company could be more efficient.

1.3. Related work

There have been a lot of studies around the topic of opinion leader identification and mining. The earliest studies go back to the 50s when all studies about centrality and influential network streams evolved.(Bavelas 1948)
With the emerging of social networks and the Web 2.0 more recent studies focus on the digital tooling to find opinion leaders. In the past years several different approaches with different goals and different methods have been developed to achieve the goal of finding opinion leaders in network structures.
The type of platform may vary from study to study, but the overall goal of all studies stays the same and nearly all approaches can proof quite good results.

Li & Gillet (2013) compared three different types of centrality and in particular closeness, betweenness and degree centrality. They used these metrics to elaborate different questions regarding certain properties of the influential users as well as investigate if there are correlations across all three centrality values. To measure those correlations they used the Spearman's Rank Correlation Coefficient, to identify the correlations between a social and an academic influence.

The social influence which is more related to social networks and other kind of social interactive boards has been elaborated by Jiang et al. (2014) and Weng et al. (2010). Both used the approach of link analysis to measure the social influence and identifying opinion leaders in the networks. They used a variation of the PageRank(Page et al. 1998) algorithm to achieve this goal. Weng et al. (2010) created an improved rank called TwitterRank which helped them to achieve their goal in not only measure the influence of a specific Twitter user with PageRank, but also a topic-sensitive influence measure. They developed a topic-sensitive influence measure because of their broad

dataset, which has been created out of Twitter users across Singapore without any topic restrictions.
Jiang et al. (2014) added an additional weight to the PageRank to improve the accuracy of the measures in their results, but there is not a huge difference between the original PageRank and the improved one. Some of the results are similar and the differences mainly vary in one rank. Another approach to identify opinion leaders in Twitter has be done by Cha et al. (2010), they compared the in-degree with mentions and retweets and figured out that the in-degree value represents popularity but misses some other notions like engagement of the users, it can be assumed that the PageRank algorithm which calculates the importance of a node creates better results than measuring the in-degree.
Another improved degree approach has been done by Ma et al. (2012), where they created an attribute index of eight degrees. It seems like this approach opens slightly better results in case of degree measurement, but they applied their metrics only on a graph of about 350 nodes and it is not proven that this is effective in a large scale network.

All approaches measure the influence of nodes within a network in a certain way, but to evaluate the users it is important to know which properties could be important to rate opinion leaders. Vollenbroek et al. (2014) used a Delphi study to figure out those properties. The Delphi study has be applied in two rounds, in the first round the participants gathered input for the second round where they needed to rate the developed influence indicators from round one. Twelve experts from social media and marketing participated in this study and the results stated that the two top indicators of influence are how often “a message is shared” and “a message has many responses”, this can be combined in one overall topic namely *engagement*.
The engagement value is such an important Key Opinion Indicator (KPI). Every page and person can easily get a lot of fans or followers – just by a first awareness and interest. But it only gets really interesting when pages succeed in the second step,too: keeping the interest and transfer it into consumer loyalty. And this shows the engagement rate. Did the content get the target group to interact. The more people interact with content, the better the Facebook algorithm is influenced. So longterm the social reach increases. So, when influencers publish content of brands and they have a great engagement rate, the more successful it will be.

The research in the related work shows that there has been an active development in the field of opinion leader mining and this provides a very good starting point to achieve the goal of finding opinion leaders within the network Instagram. While most

of these studies have a very specific problem which they try to solve around the topic opinion leader mining, this study focuses on the pure identification of those opinion leaders. Hence this study makes use of the most relevant concepts from the related work and tries to resolve the problem.

2. Relevant concepts from graph theory

This chapter will cover relevant concepts from the graph theory and in particular the topic around link analysis and how it is possible to rate nodes inside a network. Furthermore some common network effects will be described which can be observed in different types of networks.

The beginning of this chapter will describe the theory of centrality and the differences between directed and undirected graphs. Centrality plays an important role in network theory and describes how data flows through a network. This type of network theory can and has been applied to many different fields like sociology, medicine and influencer mapping.

2.1. Undirected graphs and centrality

First of all we take a look at undirected graphs which have some differences to directed graphs in terms of how calculations are applied. One can understand undirected graphs as the connection between friends within social networks. Each friend is one node inside a graph and the friendship between them is an edge connecting each other. Centrality plays an important role in network theory and is a key measurement for different types of network behaviours, for example like information flow or leadership. To understand how someone can take in the position of an influencer or leader it is important to understand how we can measure centrality and what are the basic differences.

The first researches on centrality were done by Bavelas (1948) and Bavelas (1950) at the Massachusetts Institute of Technology. These researches were based on the group structures and communication patterns within social networks. Bavelas was the first to mention the possibility, that the most central point in a network could be in the role of a leader.

We take a look at some basic undirected graph examples to understand graphs and furthermore how we can calculate centrality.
The most basic examples of graphs are the three in figure 2.1. The left one is the star

graph which visually has the most central point in the middle, the second graph is kind of a circle graph where all nodes are evenly connected and the last one is a line graph where it is difficult to tell which is the most central node in the graph.

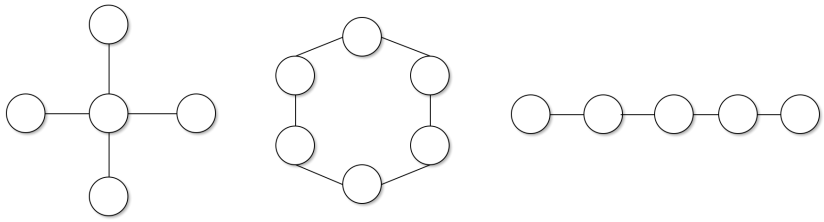

Figure 2.1.: Three different types of graphs

2.1.1. Dregree centrality

Degree centrality is the most simple calculation of centrality. The principle of degree centrality is to sum all neighbours of a given node/point, which are connected by a path. The point with the highest degree or most adjacencies is the most central point.(Freeman 1979) The basic equation for degree centrality is the following, taking a point Pk:

$$C_D(Pk) = \sum_{i=1}^{n} a(Pi, Pk) \tag{2.1}$$

Taking the star graph from figure 2.2 as example, the number of degree would be calculated as follow:

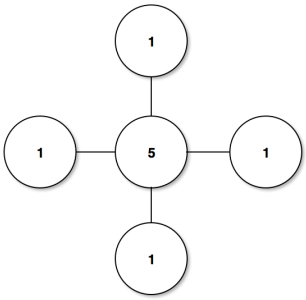

Figure 2.2.: A star graph with degree values

2.1.2. Closeness centrality

The concept of closeness centrality was first defined by Bavelas (1948). Closeness centrality calculates the shortest path for each node to all other nodes within a network, which means that they are highly well connected within their network.
Nodes with a high closeness centrality have a very short communication path within their network but this can be split upon clusters.
An improved measurement of closeness centrality index has been developed by Sabidussi (1966) as:

$$C_C(Pi)^{-1} = \sum_{k=1}^{n} d(Pi, Pk) \tag{2.2}$$

where *d(Pi,Pk)* is the sum of edges linking *Pi* and *Pk*. Nodes with high closeness centrality have the effect that messages or data is fastly distributed into the network extinguished from this node.

2.1.3. Betweenness centrality

The measurement of betweenness centrality was defined by Linton Freeman in 1977. Nodes with a high betweenness centrality act as bridges in information flow and all information is routed through this node.
These nodes can defuse the information flow and can act as a gatekeeper within a graph and can play an important role. The measurement of betweenness for point i is described as:

$$C_B(i) = \sum_{s \neq t \neq i} \frac{\sigma_{st}(i)}{\sigma_{st}} \tag{2.3}$$

where the total number of shortest path going from any node s to any node t through node i and divide it by the total number of short path between node s and node t.

2.1.4. Eigenvector centrality

The eigenvector centrality calculates the importance of a node based on the importance of its adjacencies nodes. This calculation can be applied on graphs with high or strong connectivity. It was first developed by Bonacich (1972).
While the degree centrality focuses on counting the number of adjacent nodes, the eigenvector centrality calculates the nodes importance by iterating recursively over its adjacencies.
Considering a simple example where a person is connected with Germany's leader Angela Merkel and Angela Merkel has a high importance for many others, the person as

a node gets a higher value as well as nodes connected to this person.

The eigenvector centrality is described as:

$$v_i = \frac{1}{\lambda} \sum_j A_{ij} v_j \tag{2.4}$$

where v_i is the node importance of node i and it is the sum of the importance of the neighbours. To converge within the calculation we need a damping factor to normalize the iteration, which is described with

$$\frac{1}{\lambda} \tag{2.5}$$

which is the largest eigenvalue of A, otherwise the importance measures would grow infinitely large.

This is how the importance of a node in a undirected graph can be measured. The PageRank algorithm makes use of the eigenvector centrality as well, but it was developed for calculating importance of a node for directed graphs.Page et al. (1998)
Later on we will discuss how PageRank is calculated and how it helps to solve the problem in finding important nodes in a given network.

2.2. Directed graphs, in-degree, out-degree and prestige

Directed graphs show how nodes are connected with each other in form of how data flows through the network. One node is connected in a specific direction with other nodes. This normally occurs if one only has one communication path and the communication is not per se a bi-directional communication or in mathematical description: taking an adjacent matrix in undirected graphs if node A is adjacent of node B, then B is adjacent of A as well. This is not the case for directed graphs where A and B must not necessarily be connected.
The world wide web is a simple example of how an directed graph works, it is not necessary important that every website which links to a specific website receives a backlink from this page.

In-degree: In directed graphs one node has a given amount of other nodes pointing to it, like node A in 2.3 with three edges pointing to it. The amount of directed links or edges is called in-degree for the node and therefore node A has an in-degree of three.

Out-degree: The out-degree is measured in the links pointing from a given node

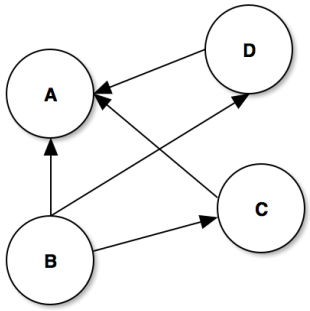

Figure 2.3.: A directed graph of four nodes

to a specific other. In 2.3 node B has an out-degree of two.

Prestige: Centrality and prestige both give information about the importance or position of a node within a network. Centrality doesn't care about the direction of a path if a node is connected with another one it gains centrality in different centrality types as described earlier. In contrast a node gets higher prestige if it has a higher amount of in-degree than out-degree. And it is important that prestige can also be negative. The PageRank algorithm for example calculates the prestige for a node within a network. Page et al. (1998)

2.2.1. PageRank algorithm

The PageRank algorithm is a very common algorithm for link analysis and was developed by Lary Page and Sergey Brin in 1998. This algorithm is based on the Eigenvector algorithm and helps to identify important nodes in a network. The PageRank algorithm was developed for the web and works with directed graphs, in a network every node starts with an equal PageRank and from there the algorithm iterates over every node and distributes the new calculated PageRank.

The basic update rule would be described as follow:

> *"Each page divides its current PageRank equally across its out-going links, and passes these equal shares to the pages it points to. (If a page has no out-going links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives."* (Easley & Kleinberg 2010, p. 407)

PageRank scales very well with large scaled graphs and the convergence takes about 50 iterations on average. A problem for this algorithm are dangling links which are nodes inside the network which receive inbound links but have no outbound links. The issues with these nodes is that is not clear where their weight should be distributed to, but they can be deleted before computation as they don't affect the PageRank of any others nodes and later on they can be added again. Page et al. (1998)

This algorithm has been developed for the web and for directed graphs as the world wide web is. Many social networks are directed graphs as well for example Twitter or Instagram, in both of these networks it is not important or necessary to follow or connect with a user. Thus the PageRank algorithm can also be applied to social networks to rate connections inside the graph. The mathematical definition of the PageRank algorithm can be described as follows:

$$PR(u) = \frac{1-d}{N} + d \sum_{v \in B_u} \frac{PR(v)}{L_v} \tag{2.6}$$

where *PR(u)* is the PageRank for a node and B_u is the set of page pointing to u and let L_u be the number of nodes u points to.
The value d will be used for normalisation and to calculate the probability value, this value defines the random surfer model and simulates the probability that a user randomly resets the surfing and starts at a new node. The so called damping factor is commonly set to 0.85.

2.3. Network effects

Taking the previous mentioned foundation theories, further valuable information can be derived. This section explains possible network situations.

2.3.1. Information cascades

In social networks it is possible that people within the network influence each other, where some influence more and some less. There are various different positions where people influence each others.
For example if it comes to choosing a restaurant, buying technological goods, political decisions and so on, it is possible that people in a network influence all of the others within a cascade. In some situations this influence depends on the personal benefit one gets through following decisions, which is called **Direct-Benefits Effect**.(Easley & Kleinberg 2010) This effect perfectly fits into the marketing world as described in

the problem description. The decision of one person could be used to influence others in their decisions.

In case of the intention to buy a gaming console a person would likely take a look at what type of console is the most used in his/her network. The decision the person makes has direct affects on his/her benefits because with the same device this person is able to play with his/her friends.
We need to keep in mind that prior to the benefit decision there could have been a decision cascade of the network before the person gets that information.

Easley & Kleinberg (2010) defined an example where decisions happen in a sequence and every node knows the previous decisions.
Taking a node A which decides first, then node B has the information about A's decision and his own. Either B accepts A's decision or chooses his own. When it comes to a third node C, then C has two information about A's and B's decision. If both decisions are different C would most likely choose his own decision, but if B accepted A's decision, then C would have two decisions of the same type. And we assume that C is likely going to choose their decisions, which means a cascade begins if the difference between rejections and acceptance reaches two. (Easley & Kleinberg 2010)

This sequence of information influences others in how they form their own decisions and may influence others. What leads to the possibility of opinion leaders within social networks, where opinion leaders have a strong influence among others.

2.3.2. The popularity effect

Popularity is an effect where some people get more public attention than others and taking a influential position in a network. This can also be applied to the web, for example a website like Wikipedia has a high amount of popularity. In the early stages of the web there have been assumptions on how this effect could be measured. A basic example is, a website has N *in-links* pointing to it, the higher the amount of N is, the higher is the popularity.(Easley & Kleinberg 2010)

3. Data mining

Data mining is an important part of the study to use information collected from a specific social network. This chapter will describe which network will be used for data mining and which data gets collected. Another part is the content crawler, how it is set up, structured and how it collects data.

3.1. Selecting the network

It is possible to find opinion leaders in nearly any social network and there are currently a lot of active social networks. When it comes to choosing the right one, it is somehow the same process as it is for an opinion leader. Of course, there are many opinion leaders which are present in different social networks, but the nature of some forces topic restrictions and the present of opinion leader in that social network.

For instance, the social networks Instagram and Pinterest mainly have image heavy content in comparison to Twitter, for example. And taking the topic of web technologies or web development, most of the content is very text heavy, which leads to the fact that nearly all opinion leaders in that specific field are more on platforms like Twitter. And for example the topics of food, vegan food, cooking and so on tend to have a much higher visual content. Many users take pictures of things they have cooked or ingredients they have used. It is much more likely, that one will find more opinion leaders for food on Instagram than on Twitter.

For the scope of this study the topic of food will be chosen as the research field. Therefore it should be a social network with its focus on image sharing. Instagram is a very popular social network for sharing images and videos. One main difference between Instagram and Pinterest is that on Instagram the intention for the user is to publish own images or videos, whereas on Pinterest it is more like a bulletin board which curates all kinds of links and images. In addition Instagram has a much broader target group in comparison to Pinterest, but as mentioned earlier, it highly depends on the topic one will focus on in finding key opinion leaders. Pinterest for example is

a very popular platform for the topic of "Do it Yourself (DIY)". From this perspective Instagram is the right platform for the scope of this study.

3.1.1. Instagram anatomy

Instagram is a simple social network which allows users to share images taken with their smart-phone from the Instagram application. Instagram gained a lot of popularity due to their application mechanisms which allow to apply simple image filters to the photos, which makes many photos visually attractive. Besides the photo sharing component users can obviously connect with each other like in every other social network.
The connection mechanism acts similar to that of Twitter. Each user can follow another user within the network but if one follows another user the user who is followed does not need to follow back. This creates a directed graph between the users.
After following a person the user is able to see all photos of this person in his personal news stream, which aggregates all images of the followed people. It is also possible to see activities like "commenting" or "liking" an image in separate streams.
A basic user application flow would be: Scrolling through his/her Instagram stream or so to say feed and "commenting" or "liking" images he/she favours. Images with a high amount of likes or comments get the status of popular images in the network.

It is possible to search the network through the application or web interface for different properties. One can search for people, tags or places. The option "people" searches the whole userbase for the given search key, the option "places" shows all images which are marked with the given location and the option "tags" are the properties to categorise images.
Every user can tag his/her images with any word they want. When publishing an image it is possible to create a caption for the image and within the caption the user starts tagging by typing a hashtag (#) and directly afterwards the word as a category. There are some very common hashtags which are used within the network, for example *#nofilter* (states that the photo was published without any filter applied) or *#tbt* (this is a popular hashtag and abbreviation for throw back Thursday, where people post images from the past).
With these tags it is possible to create a community on Instagram for certain topics as they categorise the image and users can search for their favourite topics. For example the hashtags *#veganfood* and *#cycling* both create a category for a very specific topic. One for images around vegan food and the other one for riding bicycles.

A basic Instagram post has the following components which are visible to the users and which can be used to interact with the post:

Table 3.1.: Instagram post structure

Image	The image a user shares with other in his/her network.
Caption	A simple description for the shared image.
Tags	A list of tags associated with the image, the tags are displayed at the end of the caption.
Mentions	Mentions are links to other users, they are invoked with a @-sign following a user name.
Timestamp	Shows the passed time since the post was published.
Comments	The list of user comments.
Likes	Shows the amount of likes the post received.

Figure 3.1 shows an example of an Instagram post.

Figure 3.1.: Example of an Instagram post

3.2. Data collection

To analyse the structure of Instagram and to find potential key opinion leaders it is important to scrape posts and user data from Instagram. This can be achieved by the open API provided by Instagram. The API allows to scrape data about most recent published images filtered by a tag name, information about a user and their profile, if it is a public profile as well as detailed information about posts like users who commented or liked the post. There are some more endpoints which can be used to collect

data but not all of them are important for this study.

To access the API a so called client id is necessary and can be created via the Instagram developer board and an Instagram account is needed to create a client. A client could be any project which needs access to the API, once created, a client id and client secret can be accessed and used to submit certain requests against the API. Instagram provides a RESTful API with different endpoints which can be accessed via web or with any programming language one is familiar with. Any programming language should be applicable for this concept, which has the potential of crawling data from a given web resource.
All different endpoints are listed within the platform documentation as well as different rates for specific endpoints and other restrictions.[1] Since it is a RESTful API, each endpoint may support four different HTTP methods, which are GET, POST, PUT and DELETE. These request routes could also be defined as read, create, update and delete. Some endpoints restrict certain methods for example deleting an object from the given endpoint.
The content crawler only needs the option to read a certain endpoint, this happens through the HTTP GET method. If an endpoint is called via a GET request a JSON payload will be returned containing the data. A basic JSON payload will look like the following:

```
{
    "meta": {
        "code": 200
    },
    "data": {
        ...
    },
    "pagination": {
        "next_url": "https://api.instagram.com/v1/tags/food
           /media/recent?client_id=47
           a0479900504cb3ab4a1f626d174d2d&max_id=13872296",
        "next_max_id": "13872296"
    }
}
```

[1] Instagram Platform Documentation: `https://www.instagram.com/developer/`

The **meta** object contains information about the status of the request. If the request was successful, it will return the HTTP status code 200 for success, if for example an authentication error occurs the meta object will return the status code 400 and additional information about the error.
The **data** object holds all data of the content of an object for the specific endpoint. If the endpoint for recent posts gets called the data object will contain a maximum of the last twenty posts if available.
To load more posts, if there are more than twenty, the **pagination** object will contain a **next_url** property which can be called to access the next twenty posts in a chronological way.
The API has some limitations which need to be addressed if content is crawled. One important limitation is the API rate limit, which is 5000 requests per hour which should not be exceeded otherwise the specific application will be blocked and no requests can be submitted until the hour ends. Another limitation is that not all data will be in every response, for example if posts for a specific tag are requested, only four comments and four likes are listed with full content in the initial response. There is only a number indicator if there are more of them. Thus, if one needs more data about likes and comments another endpoint needs to be called which is exclusively for either of the two properties.

3.2.1. Which data needs to be collected

To define the data which is important and needs to be collected, we need to define a node for the network and decide which things are important.
In section 2.2 the in-degree and out-degree have been explained, both these values are one of the main keys to rate the node.
There are three different values which can act as in-degree and out-degree. The first one would be the follow of a user, the second would be commenting on a post and the third one liking a post.

The following of a user is a good identifier for in-degrees but it does not necessarily give evidence about interaction for this connection. Just because the follow exists does not necessarily state, that the user who follows consumes the content of the other user.
On the contrary the liking and commenting on a post forces a user interaction with the content. This means a stronger connection and both these values are preferred to be collected. Commenting a post is a bit different in comparison to liking, since it is possible to mention another user in a comment which creates a new link. It seems like

commenting is therefore stronger than liking, but it is often the case that there are way more likes than comments on a post. For consistency and ease of use the focus will be on the likes of posts. All likes to a post are counted as in-links or in-degree and all likes from a user to a post are counted as out-links or out-degree.
To summarise, post data for a specific tag and all likes for each post need to be collected in order to fulfil the requirements. The complete post data gets collected if it needs to be analysed later on.

The topic for which the data will be crawled is food, by this time writing there are about 186 million posts on Instagram regarding to this topic. This is a lot of data and would take a long time to process, therefore a sub-topic from the food field is selected to get a more valuable result in a shorter time. Vegan food will be the sub-topic. For this field one can find about 2.2 million posts for the hashtag *#veganfood*. Choosing a sub-topic for specific field helps to filter out some non-valuable content, since Instagram is a huge social network with a large amount of users and for many topics there are posts which are not valuable or have less quality regarding visual appeal and additional benefit.

3.2.2. How the data will be collected

The data will be collected by creating an automated crawler which collects data from the Instagram API. The crawler continuously makes HTTP requests against the API Endpoints and processes the JSON response and stores the data in a database.
The crawler will be written in the programming language PHP with the use of the popular Laravel framework. Laravel sits on top of PHP and has some very useful functions, which help to develop a fast and stable product.
Laravel makes it very easy to create additional logic for processing data or monitoring, for example like adding a facility to create queue-able jobs with the help of different queueing services. This helps to create an automated persistent crawler with a lot of reliability, reducing resources and keeping the crawler safe from reaching the API rate limits defined by Instagram.
To run the crawler a server will be used where the crawler can run without any interruption, but it could be run on a local device as well.
When the crawler receives a payload from the crawled URL, the posts will be iterated and checked if they have more than four likes since then another requests needs to be done to get all likes. Only posts with more than four likes are considered to be valuable for the evaluation, because these users are likely stronger nodes in the network.

3.3. Infrastructure architecture

The infrastructure architecture consists of three different components, the database system, the crawler and the underlying web server and a queueing service to constantly run the crawler job.
To run the crawler a server is used with the following specifications:
Apache is used as web server to run the crawler with PHP as FPM application for better performance. To store the data from the Instagram API a MySQL server is used. For processing automated jobs Amazon Web Services(AWS) are used and in particular the Amazon Simple Queueing Service, which provides an simple interface to run asynchronous jobs, how this exactly works will be described later on. Figure 3.2 shows the complete architecture diagram of the whole system.

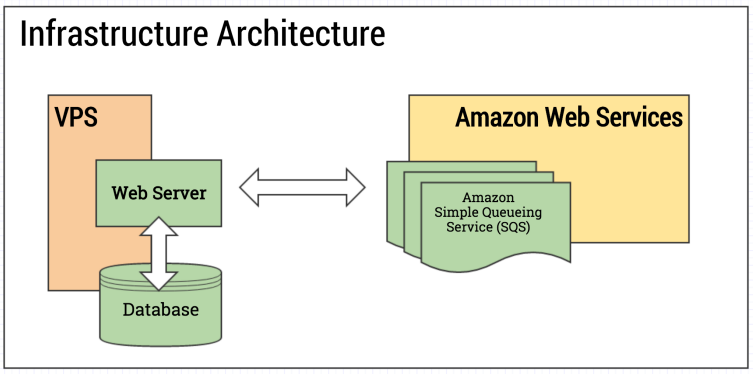

Figure 3.2.: A diagram of the underlying infrastructure

3.3.1. Database design

The database stores all data which gets collected through the crawler, it consists of two tables. One table stores all content from the collected posts to have all the information in place if it is needed for further research. And the other table is the main table where all user information is stored, this is the user identifier the corresponding inbound and outbound links.

Figure 3.3 shows the database schema which is used to store the data collected by the crawler. Both tables have a unique index ID to query the specific record as well as timestamps for updated and created times, this helps to see how long it takes from the first to the last post crawled. The user table is the more important table from which the user graph will be built later on. This table has three columns to store data. The user_id is the unique identifier from Instagram, each user has a user id from which the

user can be queried through the API. The likes column stores all likes the user received for his posts, these likes are the inbound links in graph language. And the last column are the outbound links, they will be created later on from the inbound links, they are needed to create the user graph to apply algorithms and run statistics.

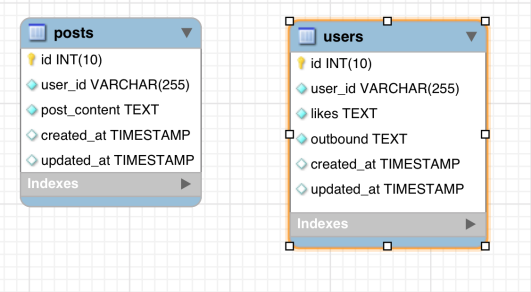

Figure 3.3.: Database schema to store crawled data

3.3.2. Crawler Architecture

The API crawler gathers all the data to a specific tag from Instagram. Once the crawler is started it runs until it is stopped manually or specific thresholds are reached. There are two values which can stop the crawler if they occur. Firstly when the amount of users in the user database reaches 250,000 and secondly if the time a post was published is two weeks in the past from the time the crawler was started.
When the crawler starts it creates a new job with an URL pointing to the specific API endpoint. This job is queued to Amazon SQS and gets called when the crawler is ready. A background job continuously checks if a job is in the queue and if there are free resources the job gets executed.
To start the crawler an initial URL is used as a starting point and the following URL will be used:
https://api.instagram.com/v1/tags/[TAG-NAME]/media/recent?client_id=[CLIENT-ID]

The URL contains the following parts:

Table 3.2.: Parts of an API request URL

API Base URL	https://api.instagram.com/v1/
API Endpoint	/tags/TAG-NAME - the second parameter defines the tag which gets crawled
Additional Endpoint Options	/media/recent - this describes all posts for the given tag in chronological order
Client ID	This is a specific identifier which allows the use of the API and can be created in the Instagram Developer back end

A crawler class processes the job by creating a cURL request against the endpoint and receives the data from Instagram, as described earlier the endpoints returns a payload containing twenty posts as JSON.
The program iterates through these posts and saves them to a database table. Afterwards it checks if the post has more than four likes, if this condition is true another cURL request is started to gather all user ids from the likes endpoint which is:
https://api.instagram.com/v1/media/[POST-ID]/likes?client_id=[CLIENT-ID]
From the returning JSON payload an array will be created which stores all user ids from this payload.
After this process another validation checks if the user which created the post exists in the user table, if this is not the case the user record will be created and the previous gathered likes are stored with it. If the user already exists, only the likes will be updated and inserted into the database. This creates two database tables which grow over time and contain all post data as well as user ids which are nodes in the networks and the likes which are all inbound links for this node.

If all twenty posts are completely processed, the next job will be queued with a new URL which is returned within the payload from Instagram. The job gets a delay of one second, which helps limit the API rate. Figure 3.4 shows a typical crawler process. The service runs autonomously and needs to have some kind of monitoring in order to be aware of the current status or if any problems occur. To monitor the crawler, a simple messaging service has been implemented to do this job. Slack is a modern messaging tool and is built on the very popular application layer protocol IRC, with helpful tools like real time messaging API. The API of Slack can help to develop certain applications which need communication implementation or in this case serve

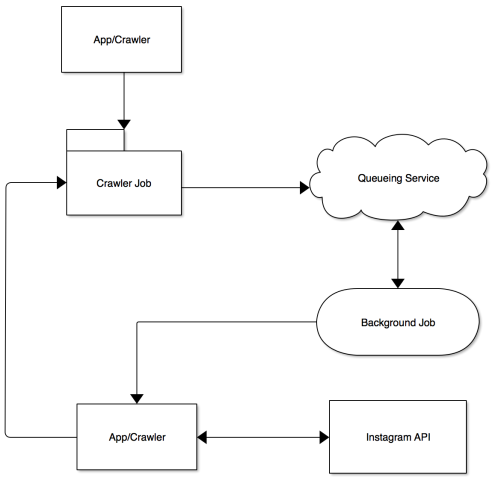

Figure 3.4.: A typical application flow for the crawler architecture

as a simple broadcasting system for monitor messages.
Some functions of the crawler can throw exceptions which cause harm to the application and may cancel the crawler. These exceptions get caught and sent via a message to a specific Slack channel. The advantage of this, is that one can receive a notification to a mobile device. This creates a very simple and cheap monitoring solution. Additionally to the failure monitoring the current status of the crawled data is sent to the Slack channel every hour to keep track of how much data has been crawled to detect possible anomalies.

3.3.3. Amazon Simple Queueing Service

Amazon SQS is a cloud-based service of the popular cloud service Amazon Web Services. It can be used to distribute messages across a large amount of receivers. This service can be used to manage and queue jobs which helps to reduce resources as well as for running them in the background.
The application can send a message to Amazon SQS which is then stored in the SQS queue. If a message is still in the queue, the new message gets attached to the queue and will be processed sequentially.

To receive the messages from SQS, a background job runs on the server which continuously checks the SQS for new messages. Whenever a message is in the queue the background job receives the message and it gets processed in the crawler application. The messages which get queued contain the pagination URL from the previous Instagram payload, when the message gets processed by the background job a new crawler

job will be launched with the next URL to process more posts. This mechanism proceeds as long as the crawler's threshold conditions have not been reached.

4. Data processing and analysis

In this chapter the method to build the network graph will be described and a discussion about different tools and libraries to analyse the graph is provided.
Furthermore it will be explained how the selected tools are used to apply algorithms to analyse the network graph and filter out results. Additionally, the set of selected algorithms will be discussed.
The data resulting from the analysis are discussed at the end of the chapter, containing conspicuousness found after applying the algorithms. The value of the results will be evaluated in the next chapter.

4.1. Analysis method

There are different methods and approaches to identify specific people within a network, besides the graph theory, which is mostly used to identify how nodes are connected in a network, there are other methods which could be considered as well to find opinion leaders.
To identify the right solution for this analysis, three different methods are discussed. As mentioned before the graph theory can identify the connection between nodes and how strongly the node is connected or how important one node is. Additionally, graph theory can provide an assumption of how data flows through the network and which nodes may act as gatekeepers.
Another method would be a sentiment analysis or also known as opinion mining, which makes use of natural language analysis to extract written language snippets and rate them. Commonly this rating is defined in three steps: positive, negative and neutral. Duan et al. (2014) applied this approach on a stock message board to identify opinion leaders. The stock messaging board is more like a bulletin board with authors and commenters, this makes it very text heavy. For a text heavy social network these method seems like a good approach to identify opinion leaders. Duan et al. (2014) compared their method against PageRank results and they received quite good results. Since Instagram is a social network which serves the purpose to share images, this method will not be very appropriate as the users are also attracted or influenced visually and not only by text.

A completely different method which could fit into Instagram's nature, would be an image analysis approach. Khosla et al. (2014) describe a method to measure image popularity and the prediction of image popularity. If a sample set of images exists which are visually attractive, the images from Instagram can be crawled and the method can be applied to measure the popularity of each image. This approach is interesting for a social network like Instagram with a heavy image focus, but the problem about an image analysis is that it lacks the interaction between the users.
It is important that users interact or engage with key opinion leaders, otherwise they are not as interesting.

From this point of view, the most appropriate method is the graph theory which makes it possible to identify important nodes and measure the engagement and interaction between nodes. Later in this chapter the applied algorithms will be discussed.

4.2. Building the network graph

The crawler collected data for two weeks in the past from the starting point and gathered 93,756 posts with all their data for the tag "#veganfood". The user table derived from this has a total amount of 26,754 entries. The user table now contains all user ids and a column with all likes (which correspond to the inbound links) of each user received for their posts.
This data can serve as a starting point for further analysis. In order to work with the extracted data in more detail this data has to be enhanced with user specific values (e.g. the user specific outbound links). With these data enhancements a meaningful network graph can be created.

The enhancing data collection of all user specific outbound links can be conducted by iterating over every user and create another request for receiving the inbound links against the user base. The pseudo code for this would be:

```
1 foreach user in users
2     foreach inbound in user.inbound
3         if inbound is in users
4             users.inbound += user.id
5         endif
6     endforeach
7 endforeach
```

After this process every user has his outbound links in another column. Now it is possible to create a graph from the data to apply various algorithms.

There are many solutions in different programming languages to do social network analysis and different tools help to accomplish this task. The package iGraph[1] is a fairly popular open source graph analysis library developed in C, but is available in R and Python as well. The advantage of using open source tools is that there are a lot of people contributing to those tools and those people can verify the algorithm which are implemented in them. As stated before iGraph is only fairly popular, at the time writing it has close to 400 stars on Github and 16 people are contributing to it.
In comparison another open source graph and network theory tool called NetworkX[2] is exclusively available in Python an has over 2000 stars with currently 133 contributors. It is a well documented library with huge functionality and provides the most important things to analyse graphs. NetworkX makes it also possible to plot the graphs directly and create a visual analyse of networks, however this can get very slow in large networks. Therefore Gephi[3] can be used to support visualisation. This tool is used by a lot of data scientists to apply statistics, algorithm and identify network structures. When it comes to visualisation of graphs and networks main focus lies within the performance and time factors when choosing the right tool. Alternatives to Gephi are for example Cytoscape[4] and Graphviz[5] which is a command line tool and can be integrated with NetworkX, both of these tools are open source as well. For the size of this project Gephi offered the proper solution to handle the collected dataset. Differences in results across various tools could be elaborated in the future.
Using well established tools from the open source community reduce the possible failure compared to writing all algorithms and visualisation from scratch. The advantage of the open source community is similar to peer reviews in the academic sphere, where more people can verify that the solution is correct.

The open source Python library NetworkX is a very powerful tool which can accomplish certain network theory tasks. This project can make use of this library to process data, run algorithms or create GraphML files which are in a XML graph format so that the graph can be imported into other tools like Gephi.
Python provides various data types to store the collected data. In this case a Python

[1] `https://github.com/igraph/igraph`
[2] `https://networkx.github.io/`
[3] `https://gephi.org/`
[4] `http://www.cytoscape.org/`
[5] `http://www.graphviz.org/`

dictionary is used to work with the data from the database. Large sets of data are very resource heavy and take a long time to process therefore a subset of the data will be used and some more of the less important nodes will be filtered out right away.
Less important nodes are determined by the following attributes. There are several records in the dataset which contain no outbound links, these records will be stripped out during the export process. All records with more than one outbound link will be written into the Python dictionary. This will create a subset of 13,754 nodes inside the dataset.

With the created Python dictionary it is now possible to import it into a Python script and create a graph with NetworkX. This happens by adding all dictionary keys as nodes and afterwards iterating over the outbound links to create the edges to the other nodes. When this process has finished certain other possible actions can be applied, like calculating different centralities or applying many other graph algorithms. Instead of using NetworkX as a tool for further analysis Gephi will be used which is an open source graph visualisation and analysis tool and is a well suited tool since it has all needed graph analysing algorithms built-in. From NetworkX a GraphML file will be exported that then gets imported into Gephi. After importing the graph Gephi counts a total node count of 26016 nodes which results from the creation of the graph in NetworkX, by creating an edge for two specific nodes, both nodes are added to the overall graph. Gephi makes it possible to remove these nodes again by applying a topology filter for the out-degree and remove all nodes again with less than one outbound link and all edges, which are self loops, get removed as well. This creates the starting point for the data processing and analysis.

4.3. Applying algorithms

At the start of applying the evaluating algorithms the graph consisted of 13,274 nodes and 498,172 edges.
There are several different statistics which can be applied to the graph directly within Gephi. First of all the graph diameter will be calculated which generates the average path length, closeness centrality as well as betweenness centrality. Afterwards the PageRank algorithm gets applied. As stated in section 2.2.1 the PageRank algorithm helps to rate nodes by their importance, this helps to strip out nodes with less importance. The PageRank algorithm expects two parameters for calculation, one is the probability factor a user randomly jumps to another node and the second parameter is a value which defines the smallest size a change from an old PageRank value to a new PageRank value can have until it converges and the algorithm stops. The probability

factor has the default value of 0.85 and the convergence value will be 0.0001.
The Eigenvector centrality values the importance of nodes as well, to use this in addition it can help to get a better overview during the data analysis. And the last statistics which will we be performed calculate the average degree. This algorithms creates the values for in-degree, out-degree and of course degree.

4.4. Results

As stated earlier in this chapter, the graph theory fits best from the compared methods. To rate the dataset five different algorithms from chapter 2 are used to receive results. For the results there are five metrics which will be taken into account, these metrics are

- PageRank
- Eigenvector centrality
- Closeness centrality
- Betweenness centrality
- Degree

In chapter 2 it has been explained how these algorithms are calculated as well as for what kind of measure they are useful. The PageRank and Eigenvector algorithms calculate the importance of nodes, closeness calculates the connectivity, betweenness can calculate if a node acts as gatekeeper and if much data flows through a specific node and degree centrality calculates how central a node is inside the network.
The PageRank algorithm seems to be the important algorithm right here as it rates nodes by their importance and takes the connection between important nodes into account. This is helpful since it is also possible that opinion leaders are connected between each other and one shares his/her importance to others.
The other algorithms are used to compare the results against each other, each algorithm may possibly be used for opinion leader identification. Betweenness centrality is a interesting metric because of its possibility to identify gatekeepers, as they may decide how traffic flows through the network they are able to influence users or their node neighbours.
All algorithms will be applied on the complete graph set and result lists for each algorithm will be created.

This results in a list of 100 nodes. In these 100 nodes are 83 unique nodes, consequently there are nodes in the top twenty within two or three metrics. There are 12 nodes which are currently in two metrics and two nodes within three metrics. One of the first things one notices, is that the betweenness centrality is the metric that correlates the most with another one. Besides two nodes every node with more than one metric is in the top twenty of betweenness centrality. The results for betweenness metric can be seen in table 4.1, all nodes coloured in teal are in two metrics under the top twenty and nodes coloured in green are in three metrics under the top twenty. In the appendix all tables for each of the different metrics can be found. There are only two nodes which do not correlate with betweenness but are in two metrics, these two nodes are the number one and the number two of the top twenty PageRank results and are also number one and number two within closeness centrality.

Table 4.1.: Top twenty for betweenness centrality

Rank	**User ID**	**Betweenness Centrality**
1	228149983	0.01675175275
2	2965017327	0.01583687569
3	2257327590	0.01469924504
4	1809266129	0.01432848231
5	516447227	0.01297032692
6	243261468	0.01245180453
7	2928223248	0.01205098845
8	2289505127	0.01173518382
9	2957793201	0.01145951201
10	257407321	0.01143626078

Rank	**User ID**	**Betweenness Centrality**
11	2521327556	0.0100404406
12	2293413042	0.009935677663
13	37628499	0.009908867692
14	1655365372	0.00980436694
15	2348617947	0.009527888045
16	2404675952	0.009412891866
17	188995785	0.009397996366
18	2363137244	0.009188458379
19	2233285274	0.009090333734
20	232817675	0.00907133629

Based on the results and the top accounts more data must be collected to evaluate the results to see if it is possible to select opinion leaders from this data. As stated earlier there are 83 unique nodes, these 83 nodes form the set of users which are assumed to be potential for opinion leader selection. In order to evaluate the results and finding out if there are some opinion leaders relevant for marketing purpose, more data for each specific user will be gathered from Instagram. For this reason the API is used again to collect information about each user and the additional informations to be collected are the following properties: followed_by (amount of users which follow the user with this user ID), follows (amount of users the specific user ID follows by him/herself), media (amount of posts published) and user name. The user name is important in order to view the profile in the web later on, this helps to rate the quality of the posts and im-

ages. The complete set of results for every different metric can be seen in the appendix.

It is hard to tell what properties are important to measure the value of a user, of course this can be done by a lot of different metrics as well. For the sake of convenience, nodes with a good rate of more followed_by than follows are considered as high valued nodes. Nevertheless nodes with even amount or a low values for these properties can be identified as opinion leaders, for example if they are in a growing network and their account is new or if the media they create is visually very attractive and of good quality. It is also necessary to keep in mind that users can get very popular without a huge amount of followers because it is not important to follow a user in order to like or see a post of the user.

5. Evaluating results

The top twenty across all metrics have been extracted and there are a lot of users which have an interesting amount of followers to follows. The first thing to notice is that there are no users with a huge amount of followers except for one. During the research it was figured out that the user behaviour of Instagram changed over the last couple of months. Many popular users with a high amount of followers tend to write their image tags inside a comment of the post. This has the effect that they can be found within the Instagram application by using the search, but their posts are not part of the APIs *media/recent* endpoint. And of course this user behaviour leads to a distorted result, though it seems that there are potential key opinion leaders in the unique 83 posts.

As mentioned in the results section 4.4, the results for betweenness centrality show that nearly all nodes in this metric are present within at least two metrics. To evaluate the overall outcome the results of PageRank and betweenness centrality will be compared by their distribution to the amount of followers each user has. Figure 5.1 and 5.2 show the distribution of both metrics compared to the amount of followers a user got. It can be observed that the betweenness centrality is much more scattered than the PageRank values. Overall the PageRank only has two nodes which show an anomaly in the result, because of their high PageRank value compared to the amount of followers.
This indicates that the PageRank results are much more stable when it comes to similarities in follower counts. Furthermore the graphs show that the average amount of followers is considerably higher for PageRank, whereas a lot of users have been identified around the 10,000 follower mark. The closeness centrality results don't need to be compared since the top twenty results got all the same value and seem to be evenly connected across the network. The graphs for the distributions of the Eigenvector and degree results can be seen in the appendix B. One interesting thing to notice is that the Eigenvector distribution is very scattered as well even though it rates notes to their importance like the PageRank as well.
The PageRank results show a lot of users which have not a huge amount of followers but a very good ratio of how many people follow the user and he/she follows other

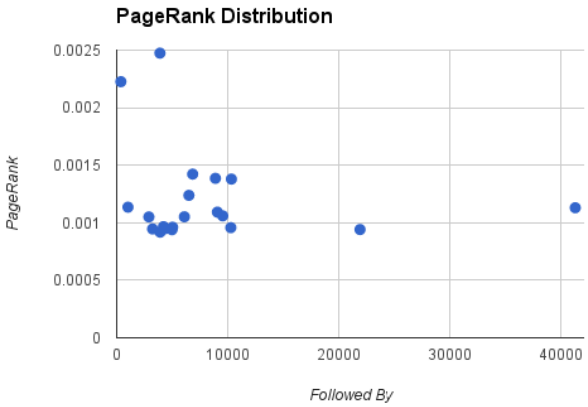

Figure 5.1.: PageRank to FollowedBy Distribution

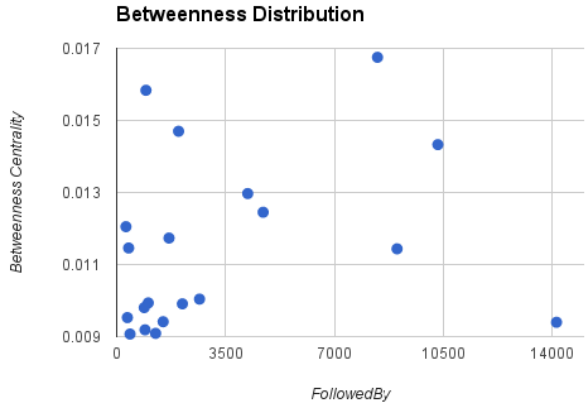

Figure 5.2.: Betweenness centrality to FollowedBy Distribution

users. It can be assumed that these users are somehow in the position of an opinion leader since there are a lot of people who consume the content these users produce. The amount of media published to the specific rank they received is important as well, because users with many published media have more likely received more inbound links during the same amount of time.

Nevertheless it seems that there are opinion leaders in the top twenty of the PageRank metric. Taking a look at the top twenty of the PageRank results, see table 5.1, one can assert that there are a some users with a really good ratio of followed-by to follows. The following users stand out of all others: sobeautifullyraw (Rank 8), elvirafrolin (Rank 15) and applesandamandas (Rank 18). The user sobeautifullyraw has over 40,000 followers which is a really good amount and inspecting the account of this user verifies

that this is a potential opinion leader. Figure 3.1 shows one post of this user and it has a very high quality and visual attractive image. By viewing the most recent posts of this user it can be identified that the user is very popular and has an average 4,000 likes and 100 comments on every post. The user elvirafrolin has not a huge amount of comments on her images but the posts are visually very attractive. In comparison to sobeautifullyraw she has way more posts. This shows the drawback of the crawled data as mentioned before, but the user still has a potential for marketing. Of course there are a lot more other users with fewer followers but with good content and a good ratio of followed-by. To get a better insight about the results a marketing expert with focus on opinion leaders will rate some of the users and state if they are of suitable for a collaboration.

Table 5.1.: Top twenty for PageRank results

Rank	FollowedBy	Follows	Media	Username
1	3892	454	952	michelnilles
2	375	281	1039	sayitloud_kampffussel
3	6833	41	521	vegansofldn
4	8882	970	1628	klean_slate
5	10320	1826	654	theplantpoweredprincess
6	6496	158	1754	thegreenedge
7	1013	252	745	afrofuzzz
8	41256	334	239	sobeautifullyraw
9	9057	4908	433	raw4zack
10	9542	2102	321	reganthevegan
11	6088	869	1284	plant_based_bigness
12	2894	7132	300	travelwithjaz
13	4210	240	339	nordic_vegan
14	5032	99	834	theturnip_
15	10266	106	780	elvirafrolin
16	4328	402	2085	sped87
17	3229	324	1259	alphablack_veganmen
18	21892	223	880	applesandamandas
19	4975	443	1303	vegannomadchick
20	3909	2598	1560	london_afro_vegan

5.1. Expert review

To evaluate the results further and see if the opinion leaders found with the PageRank algorithm are valuable an expert is consulted to additionally evaluate the results.
The chosen expert is a marketing strategist with focus on social networks and key

opinion leader marketing and over ten years of experience in the domain of marketing. The expert chose an approach to compare the results from this study against different other sources and their opinion leader recommendation. The different sources follow the experts current method to identify opinion leaders which is a manual method. One source is an article from Huffington Post[1] and the other one is a food bloggerin[2].

After collecting all the user names the expert used another tool which is called "fanpage karma"[3], this tool allows to search for certain user names across different social networks and display their statistics regarding followers, media published, engagement and many more. Figures 5.3, 5.4 and 5.5 show the different fanpage karma results for the different identified opinion leaders.
With these results the marketing expert could evaluate that the opinion leaders identified with this study have a higher engagement value than the opinion leaders identified by the other sources. The expert stated that this value is very important and that the result is extremely valuable for the engagement metric. But of course the results need further investigation and each opinion leader profile needs to be reviewed to figure out if they will be considered for marketing use.

After the last step the expert was able to figure out that there is only one opinion leader from the PageRank results which can not be considered as valuable for marketing purposes. And the result of useful opinion leaders is even compared to the other sources. The expert marked two key opinion leaders from this study as highly valuable, three from the food blog and two from Huffington Post.

[1] `http://www.huffingtonpost.com/2015/02/20/vegan-instagram-accounts-not-just-kale_n_6715422.html`
[2] `http://food.allwomenstalk.com/delicious-instagram-vegan-accounts-you-should-be-following`
[3] http://www.fanpagekarma.com/

	Follower	Following	Number of Posts	Likes	Comments	Comments and Likes	Daily Growth (absolute)	Growth rate	Percentage increase since starting point	Followers-Following-Ratio	Engagement	Post Interaction
amanda victoria | ...	23k	225	56	112k	1.7k	113k	n.a.	n.a.	n.a.	102	18%	9.0%
ELVIRA FRÖLIN	11k	107	73	54k	913	55k	n.a.	n.a.	n.a.	98	19%	7.2%
Hannah	11k	1.8k	54	33k	1.1k	34k	n.a.	n.a.	n.a.	5.9	12%	6.0%
Regan smith	9.9k	2.1k	49	25k	796	26k	n.a.	n.a.	n.a.	4.7	9.5%	5.4%
SAM ☾ MELBOURNE	51k	358	36	152k	6.8k	158k	n.a.	n.a.	n.a.	144	11%	8.7%

Figure 5.3.: Fanpage karma for KOLs from this study

	Follower	Following	Number of Posts	Likes	Comments	Comments and Likes	Daily Growth (absolute)	Growth rate	Percentage increase since starting point	Followers-Following-Ratio	Engagement	Post Interaction
Jasmine	83k	667	66	87k	1.6k	89k	n.a.	n.a.	n.a.	124	3.9%	1.7%
Kai Nora	29k	199	38	49k	549	49k	n.a.	n.a.	n.a.	144	6.2%	4.6%
kara	13k	699	0	0	0	0	n.a.	n.a.	n.a.	19	0%	0%
Nom Yourself	124k	246	39	88k	1.9k	90k	n.a.	n.a.	n.a.	504	2.6%	1.9%
Pixie, 23, UK	107k	386	33	58k	841	59k	n.a.	n.a.	n.a.	278	2.0%	1.7%

Figure 5.4.: Fanpage karma for KOLs from Huffington Post

SELECT PERFORMANCE INDICATORS | Engagement

	Follower	Following	Number of Posts	Likes	Comments	Comments and Likes	Daily Growth (absolute)	Growth rate	Percentage increase since starting point	Followers-Following-Ratio	Engagement	Post Interaction
Yovana | Snapchat:...	420k	487	60	605k	6.3k	612k	20k	5.1%	5.1%	863	5.3%	2.5%
KIM-JULIE HANSEN	113k	528	48	122k	2.6k	125k	n.a.	n.a.	n.a.	214	4.0%	2.3%
SHANTELLE ✨Atl...	22k	1.1k	49	18k	1.0k	19k	n.a.	n.a.	n.a.	20	3.2%	1.8%
Snapchat: FullyRa...	797k	463	39	514k	12k	526k	n.a.	n.a.	n.a.	1.7k	2.4%	1.7%
#NANAICECREAM	38k	183	0	0	0	0	n.a.	n.a.	n.a.	206	0%	0%

Figure 5.5.: Fanpage karma for KOLs from blogger

6. Conclusion

This study shows to some extent that it is possible to identify opinion leaders within the social network Instagram. The expert review shows that this approach identifies opinion leaders with a very high engagement rate which is very useful for marketing purpose.
During this research and analysis it has been noticed that the user behaviour of successful Instagramers or opinion leaders inside of Instagram works against this method and they are therefore not considered and not included in the graph. This results from the way these users publish their posts. They don't use tags inside their initial post instead they add all tags inside a comment to this post. This has the effect that they will be found within the Instagram application for the listed tags, but they cannot be captured through the recent tags API endpoint since the API only checks the tags used in the post description.
There are only two ways how these users could be included into the research, the first one would be they change their user behaviour back to including tags into the initial post description, but this is very unlikely. And the other option would be that Instagram improves their API to include those posts into the results, maybe the same way as they display them in their own application.
Nevertheless the results show that there are some very interesting opinion leaders which can be used for marketing purposes.
The results for the PageRank algorithm showed a solid result compared to the other used algorithms and it seems that the PageRank is a good algorithm to identify opinion leaders. Additional research could evaluate whether the other algorithms can be used to identify nodes with a specific behaviour and if it is possible to find out if these algorithms could identify different user behaviour.

Future research could combine this approach with additional approaches mentioned earlier. Sentiment analysis or image analysis could be interesting to add additional weight to each node. This could be done by creating a cloud based machine learning system.
Additionally this approach could be extended by collecting comments as well and the links which occur inside of them. Those links and comments could be added as a

connection with more weight because this takes more user interaction than a like. And since building the user graph from likes results in finding opinion leaders with a good engagement, the inclusion of comments could improve this even more.

Another future research could be conducted from a marketing perspective by questioning whether the different used algorithms create different types of opinion leader and if they differ whether they can serve different marketing purposes.

List of Figures

List of Tables

Bibliography

Bavelas 1948
Bavelas, Alex: A Mathematical Model for Group Structures. In: *Human Organization* 7 (1948), pages 16–30

Bavelas 1950
Bavelas, Alex: Communications Patterns. 22 (1950), Nr. 6, pages 725–730

Bonacich 1972
Bonacich, Phillip: Factoring and weighting approaches to status scores and clique identification. In: *The Journal of Mathematical Sociology* 2 (1972), Nr. November 2014, pages 113–120. http://dx.doi.org/10.1080/0022250X.1972.9989806. – DOI 10.1080/0022250X.1972.9989806. – ISBN 0022–250X

Cha et al. 2010
Cha, Meeyoung; Haddai, Hamed; Benevenuto, Fabricio; Gummadi, Krishna P.: Measuring User Influence in Twitter : The Million Follower Fallacy. In: *International AAAI Conference on Weblogs and Social Media* (2010), 10–17. http://dx.doi.org/10.1.1.167.192. – DOI 10.1.1.167.192. – ISBN 9781450304931

Duan et al. 2014
Duan, Jiangjiao; Zeng, Jianping; Luo, Banghui: Identification of Opinion Leaders Based on User Clustering and Sentiment Analysis. In: *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* 1 (2014), 377–383. http://dx.doi.org/10.1109/WI-IAT.2014.59. – DOI 10.1109/WI–IAT.2014.59. ISBN 978–1–4799–4143–8

Easley & Kleinberg 2010
Easley, David; Kleinberg, Jon: *Networks , Crowds , and Markets : Reasoning about a Highly Connected World*. Bd. 81. 2010. – 744 S. http://dx.doi.org/10.1017/CBO9780511761942. http://dx.doi.org/10.1017/CBO9780511761942. – ISBN 9780521195331

Freeman 1979
Freeman, L C.: Centrality in Social Networks Conceptual Clarification. In: *Social Networks* 1 (1979), Nr. 3, 215–239. http://dx.doi.org/10.1016/0378-8733(78)90021-7. – DOI 10.1016/0378–8733(78)90021–7. – ISBN 0378–8733

Jiang et al. 2014
Jiang, Lin C.; Li, Fang F.; Ge, Bin; Xiao, Wei D.; Tang, Jiu Y.; Hu, Yan L.: Detecting Opinion Leaders in Online Communities Based on an Improved PageRank Algorithm. In: *Applied Mechanics and Materials* 543-547 (2014), 3524–3527.

http://dx.doi.org/10.4028/www.scientific.net/AMM.543-547.3524. – DOI 10.4028/www.scientific.net/AMM.543–547.3524. – ISSN 1662–7482

Khosla et al. 2014
Khosla, Aditya; Das Sarma, Atish; Hamid, Raffay: What Makes an Image Popular ? In: *Proceedings of the 23rd International Conference on World Wide Web* (2014), 867—-876. http://dx.doi.org/10.1145/2566486.2567996. – DOI 10.1145/2566486.2567996. – ISBN 9781450327442

Li & Gillet 2013
Li, Na; Gillet, Denis: Identifying influential scholars in academic social media platforms. In: *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13* (2013), 608–614. http://dx.doi.org/10.1145/2492517.2492614. – DOI 10.1145/2492517.2492614. ISBN 9781450322409

Ma et al. 2012
Ma, Ning; Liu, Yijun; Tian, Ruya; Li, Qianqian: Recognition of online opinion leaders based on social network analysis. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7669 LNCS (2012), pages 483–492. http://dx.doi.org/10.1007/978-3-642-35236-2{_}48. – DOI 10.1007/978–3–642–35236–2_48. – ISBN 9783642352355

Page et al. 1998
Page, Lawrence; Brin, Sergey; Motwani, Rajeev; Winograd, Terry: 1 Introduction and Motivation 2 A Ranking for Every Page on the Web. In: *World Wide Web Internet And Web Information Systems* 54 (1998), Nr. 1999-66, 1–17. http://dx.doi.org/10.1.1.31.1768. – DOI 10.1.1.31.1768. – ISBN 9781424433803

Sabidussi 1966
Sabidussi, Gert: The centrality index of a graph. In: *Psychometrika* 31 (1966), Nr. 4, pages 581–603. http://dx.doi.org/10.1007/BF02289527. – DOI 10.1007/BF02289527. – ISBN 0033–3123

Vollenbroek et al. 2014
Vollenbroek, Wouter; De Vries, Sjoerd; Constantinides, Efthymios; Kommers, Piet: Identification of influence in social media communities. In: *International Journal of Web-based Communities* 10 (2014), Nr. 3, pages 280–297. http://dx.doi.org/10.1504/IJWBC.2014.062943. – DOI 10.1504/IJWBC.2014.062943. – ISSN 17418216

Weng et al. 2010
Weng, Jianshu; Lim, Ee-Peng; Jiang, Jing; He, Qi: Twitterrank: Finding topic-sensitive influential Twitterers. In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining* (2010), 261–270. http://dx.doi.org/10.1145/1718487.1718520. – DOI 10.1145/1718487.1718520. ISBN 9781605588896

Appendix

A. Algorithm results

The following tables are the results of the five metrics which have been applied to the graph. Each table is sorted for the specific metric and the top twenty nodes in each metric are displayed.

Table A.1.: Top twenty for PageRank results (full table)

PageRank	Closeness	Betweenness	Eigenvector	Degree	FollowedBy	Follows	Media	Username
0.002474	1	0.000002	0.438219	242	3892	454	952	michelnilles
0.002226	1	0.0000002	0.222371	127	375	281	1039	sayitloud_kampffussel
0.001423	0.364395	0.002663	0.404178	330	6833	41	521	vegansofldn
0.001386	0.366843	0.002147	0.289238	290	8882	970	1628	klean_slate
0.00138	0.468284	0.014328	0.902529	1480	10320	1826	654	theplantpoweredprincess
0.001238	0.239085	0.000396	0.402976	212	6496	158	1754	thegreenedge
0.001135	0.440027	0.009936	1	1136	1013	252	745	afrofuzzz
0.00113	0.31598	0.000602	0.22821	189	41256	334	239	sobeautifullyraw
0.001093	0.388589	0.004222	0.349964	633	9057	4908	433	raw4zack
0.00106	0.341526	0.001117	0.445333	316	9542	2102	321	reganthevegan
0.001052	0.37327	0.003175	0.428545	374	6088	869	1284	plant_based_bigness
0.001051	0.4246	0.004892	0.411749	596	2894	7132	300	travelwithjaz
0.000966	0.474662	0.01297	0.415273	1702	4210	240	339	nordic_vegan
0.000961	0.349049	0.001256	0.464119	341	5032	99	834	theturnip_
0.000957	0.31802	0.000342	0.369788	241	10266	106	780	elvirafrolin
0.000948	0.381329	0.002754	0.544241	428	4328	402	2085	sped87
0.000946	0.358121	0.001663	0.437644	313	3229	324	1259	alphablack_veganmen
0.000941	0.319127	0.00042	0.309414	228	21892	223	880	applesandamandas
0.000939	0.340656	0.001371	0.4487	268	4975	443	1303	vegannomadchick
0.000918	0.365799	0.001819	0.705459	477	3909	2598	1560	london_afro_vegan

Table A.2.: Top twenty for Closeness results (full table)

PageRank	Closeness	Betweenness	Eigenvector	Degree	FollowedBy	Follows	Media	Username
0.0024743	1	0.0000017	0.4382186	242	3892	454	952	michelnilles
0.0022263	1	0.0000002	0.2223709	127	375	281	1039	sayitloud_kampffussel
0.000081	1	0.0000729	0.0028181	5	104	55	68	piyathakur006
0.0001165	1	0.0000162	0.0226714	17	386	445	55	nutri_primazza
0.0001707	1	0.000016	0.1058489	80	925	43	242	my.juices
0.0001939	1	0.0000159	0.1220119	61	120	78	382	scb1005
0.0001206	1	0.0000148	0.0045398	7	936	252	162	irrelevantgenetics1.0
0.0002906	1	0.0000144	0.2306495	116	4801	1820	798	amrit_py
0.0000309	1	0.0000116	0.0219919	10	546	445	672	alanlyfal
0.0001104	1	0.0000089	0.0293751	20	51	21	64	makeup__plus
0.0001924	1	0.0000089	0.1419565	65	135	280	151	noam_komem7
0.0006038	1	0.0000086	0.1358408	70	5512	781	206	nutri.hitt
0.0001067	1	0.0000084	0.0351045	19	281	475	277	argital_australia
0.0001664	1	0.0000084	0.0732142	41	98	15	158	alignyourhealthlife
0.0001447	1	0.0000074	0.0578661	27	72	36	51	tiger_lily_sweets_asheville
0.0001521	1	0.0000072	0.0516854	39	166	42	450	hyfdiary
0.0005703	1	0.0000063	0.0189717	21	1544	1111	284	vegan4youbrasil
0.000111	1	0.0000059	0.0547003	23	184	824	497	damadyl
0.0002177	1	0.0000058	0.1133499	68	3154	658	301	dohnashville
0.000013	1	0.0000055	0.0074185	5	377	193	553	tacianasantos_ss

Table A.3.: Top twenty for Betweenness results (full table)

PageRank	Closeness	Betweenness	Eigenvector	Degree	FollowedBy	Follows	Media	Username
0.000799	0.509744	0.016752	0.410613	2628	8379	478	220	adam.biddle
0.000682	0.516821	0.015837	0.325524	2610	940	300	46	sitaelizabeth
0.000755	0.490713	0.014699	0.546606	2078	1988	74	239	inasveganway
0.00138	0.468284	0.014328	0.902529	1480	10320	1826	654	theplantpoweredprincess
0.000966	0.474662	0.01297	0.415273	1702	4210	240	339	nordic_vegan
0.00088	0.460355	0.012452	0.649537	1543	4706	425	1624	alittledishy
0.000536	0.471723	0.012051	0.614775	1520	297	214	131	mehralsgruenzeug
0.000476	0.518416	0.011735	0.377998	2628	1681	573	137	seemaskitchen
0.000802	0.469358	0.01146	0.853394	1530	383	256	109	fromsteaktosoya
0.000892	0.473274	0.011436	0.385243	1797	9009	355	541	veganality
0.000709	0.467332	0.01004	0.43652	1477	2661	148	111	hello.vegan
0.001135	0.440027	0.009936	1	1136	1013	252	745	afrofuzzz
0.000906	0.481358	0.009909	0.616153	1279	2111	1904	223	veganleanne
0.000381	0.513359	0.009804	0.454335	2459	885	87	102	alsylemon
0.000405	0.463563	0.009528	0.471807	1547	343	23	33	vegan333
0.000511	0.497645	0.009413	0.320145	2057	1492	1354	69	the25yearoldvegan
0.000624	0.454905	0.009398	0.46911	1372	14132	3776	2473	rockinmike
0.000519	0.499238	0.009188	0.405219	2054	911	8	39	lentilkiller
0.000343	0.466667	0.00909	0.472249	1603	1250	2862	137	vronikal_vegan
0.00032	0.502319	0.009071	0.28299	2355	428	305	79	beckyveganbaker

Table A.4.: Top twenty for Eigenvector results (full table)

PageRank	Closeness	Betweenness	Eigenvector	Degree	FollowedBy	Follows	Media	Username
0.001135	0.440027	0.009936	1	1136	1013	252	745	afrofuzzz
0.000731	0.286182	0.000199	0.942971	500	708	250	251	veganfullife
0.00138	0.468284	0.014328	0.902529	1480	10320	1826	654	theplantpoweredprincess
0.000802	0.469358	0.01146	0.853394	1530	383	256	109	fromsteaktosoya
0.000527	0.458631	0.008289	0.810051	1099	1070	199	606	creeddennis
0.000677	0.380709	0.001767	0.771671	513	2902	253	2110	glutenfreeveganfoodpervert
0.000768	0.448567	0.005565	0.767514	786	3314	34	427	setting_the_bone
0.000516	0.47921	0.007495	0.721153	1010	490	494	187	manya_food
0.000918	0.365799	0.001819	0.705459	477	3909	2598	1560	london_afro_vegan
0.000734	0.400893	0.002118	0.704139	484	1508	1100	192	vegan.in.italy
0.000682	0.336846	0.00059	0.693953	356	869	1768	712	hetface
0.000821	0.425786	0.004334	0.689202	664	4330	4583	2118	knittingopera
0.000877	0.332193	0.000494	0.677946	338	2670	998	2217	figsontoast
0.000578	0.366822	0.001303	0.655962	384	1684	206	795	carly_182
0.00088	0.460355	0.012452	0.649537	1543	4706	425	1624	alittledishy
0.000514	0.438246	0.006708	0.645725	1001	1554	429	1061	craftyearthmama
0.000401	0.410771	0.00208	0.64372	504	1602	418	1381	rachelrenelorton
0.000678	0.361679	0.001032	0.642055	406	1257	579	448	treehuggingearthling
0.000599	0.448952	0.003346	0.639473	468	690	355	401	veganelvfa
0.000504	0.466534	0.006423	0.624495	626	841	1	754	dnesjemvegan

Table A.5.: Top twenty for Degree results (full table)

PageRank	Closeness	Betweenness	Eigenvector	Degree	FollowedBy	Follows	Media	Username
0.000081	0.525611	0.003442	0.056404	2721	6191	658	711	jackyfalkenberg
0.000799	0.509744	0.016752	0.410613	2628	8379	478	220	adam.biddle
0.000476	0.518416	0.011735	0.377998	2628	1681	573	137	seemaskitchen
0.000682	0.516821	0.015837	0.325524	2610	940	300	46	sitaelizabeth
0.000381	0.513359	0.009804	0.454335	2459	885	87	102	alsylemon
0.000281	0.499562	0.008894	0.185053	2385	943	232	64	thegreenshelter
0.00032	0.502319	0.009071	0.28299	2355	428	305	79	beckyveganbaker
0.000254	0.508854	0.006812	0.203098	2234	758	1620	132	casapravas
0.000755	0.490713	0.014699	0.546606	2078	1988	74	239	inasveganway
0.000511	0.497645	0.009413	0.320145	2057	1492	1354	69	the25yearoldvegan
0.000519	0.499238	0.009188	0.405219	2054	913	8	39	lentilkiller
0.000446	0.500764	0.008791	0.466578	2049	393	272	177	veggyjulie
0.000308	0.509111	0.006884	0.202093	2039	1241	317	44	plantpowerrr
0.000338	0.505595	0.006276	0.237677	2007	835	234	75	miyvelvet
0.000189	0.504777	0.004701	0.166119	1986	1721	306	141	anciamainsta
0.000483	0.483115	0.008584	0.242867	1958	2574	2961	59	fredrik_litekitchen
0.000398	0.490713	0.006071	0.1939	1852	2926	739	29	laveganfoodshare
0.000482	0.478545	0.008364	0.256538	1826	1498	59	69	clara_foodie
0.000181	0.509447	0.004465	0.130885	1824	2271	147	35	deliciabale
0.000179	0.496834	0.002923	0.102256	1809	1133	203	93	livelovesmile

B. Distribution visualisation

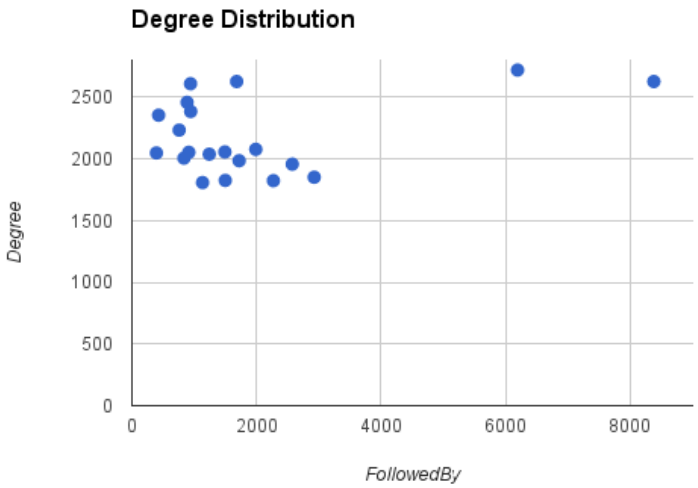

Figure B.1.: Degree to FollowedBy Distribution

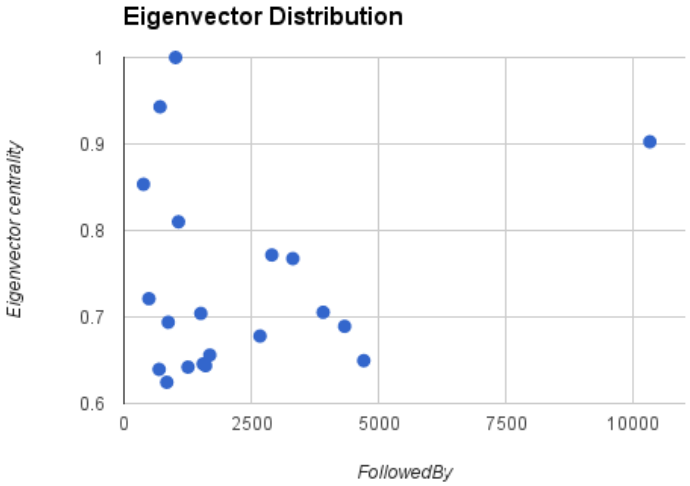

Figure B.2.: Eigenvector to FollowedBy Distribution

C. Code Repository

The used code to crawl the data and the used data set to use in Gephi are stored in the following Github repository:
`https://github.com/sourcecube/social-network-analysis`

There are two folders within the project, one which contains the code for crawling data from Instagram and the other is a small Python script to create a GraphML file or run algorithms on the graph. The used GraphML file is located within the *networkx* folder.

Declaration

I hereby declare that this master thesis was independently composed and authored by myself.

All content and ideas drawn directly or indirectly from external sources are indicated as such. All sources and materials that have been used are referred to in this thesis.

The thesis has not been submitted to any other examining body and has not been published.

Cologne, 09.04.2016

Christopher Egger